IN THE CLAIMS

Please amend the claims as follows:

1. (original) A method of providing a digital signal processing function f to an executing device in an obfuscated form; the function f including a function cascade including a plurality of signal processing functions f_i, $1 \le i \le N$, for processing a digital signal input x to yield a digital signal output (for example, $FC_1(x) \equiv f_N \circ \cdots \circ f_1(x)$), the method including:

selecting a set of $2N$ invertible permutations p_i, $1 \le i \le 2N$;

calculating a set of N functions g_i, where g_i is functionally equivalent to $p_{2i}^{-1} \circ f_i \circ p_{2i-1}$, for $1 \le i \le N$;

calculating a set of N-1 functions h_i, where h_i is functionally equivalent to $p_{2i-1}^{-1} \circ p_{2i-2}$, for $2 \le i \le N$;

equipping the executing device with an execution device function cascade that includes $y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \ldots \circ y_1$, where $y_1,\ldots,y_N$ are function parameters $\left(\text{for example, } ED_1(y_1,\ldots,y_N) \equiv y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \ldots \circ y_1\right)$,

providing the functions $g_1,\ldots,g_N$ to the executing device; and

in the executing device, applying the execution device function cascade to the functions $g_1,\ldots,g_N$ $\left(\text{for example, } ED_1(g_1,\ldots,g_N)\right)$.

2. (original) A method of providing a digital signal processing function f as claimed in claim 1, wherein the execution device function cascade includes $y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \ldots \circ y_1 \circ p_1^{-1}$ $\left(\text{for example, } ED_2(y_1,\ldots,y_N) \equiv y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \ldots \circ y_1 \circ p_1^{-1}\right)$.

3. (original) A method of providing a digital signal processing function f as claimed in claim 1, wherein the function cascade starts with a further signal processing function f_0 $(\text{for example, } FC_2(x) \equiv f_N \circ \cdots \circ f_1 \circ f_0(x))$ and the execution device function cascade includes $y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \ldots \circ y_1 \circ S_1$ $\left(\text{for example, } ED_3(y_1,\ldots,y_N) \equiv y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \ldots \circ y_1 \circ S_1\right)$, where S_1 is functionally equivalent to $p_1^{-1} \circ f_0$.

4. (original) A method of providing a digital signal processing function f as claimed in claim 1, wherein the execution device function cascade includes $p_{2N} \circ y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \ldots \circ y_1$ (for example $ED_4(y_1,\ldots,y_N) \equiv p_{2N} \circ y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \ldots \circ y_1$

5. (original) A method of providing a digital signal processing function f as claimed in claim 1, wherein the function cascade ends with a further signal processing function f_{N+1},

(for example, $FC_3(x) \equiv f_{N+1} \circ f_N \circ \cdots \circ f_1(x)$) and the execution device function cascade includes

$S_2 \circ y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \ldots \circ y_1$ $\left(\text{for example, } ED_5(y_1,\ldots,y_N) \equiv S_2 \circ y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \ldots \circ y_1\right)$, where S_2 is functionally equivalent to $f_{N+1} \circ p_{2N}$.

6. (original) A method of providing a digital signal processing function *f* as claimed in claim 1, including obtaining a unique identity of the executing device and/or user of the executing device; the set and/or sequence of $2N$ invertible permutations p_i being unique for the obtained identity.

7. (original) A method as claimed in claim 1, wherein the step of equipping the executing device with the execution device function cascade includes providing the execution device function cascade embedded in a software program for execution by a processor in the executing device.

8. (original) A method as claimed in claim 7, wherein the step of providing the functions $g_1,\ldots,g_N$ to the executing device includes providing the functions $g_1,\ldots,g_N$ in the form of a plug-in for the program.

9. (original) A method as claimed in claim 7, wherein the step of providing the functions $g_1,\ldots,g_N$ to the executing device includes embedding the functions $g_1,\ldots,g_N$ in the software program by applying the execution device function cascade to the function parameters $g_1,\ldots,g_N$.

10. (original) A computer program product operative to cause a processor in an execution device to execute a digital signal processing function f including a function cascade including a plurality of signal processing functions f_i, where $1 \le i \le N$, for processing a digital signal input x to yield a digital signal output (for example, $FC_1(x) \equiv f_N \circ \cdots \circ f_1(x)$), by:

loading an execution device function cascade that includes $y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \ldots \circ y_1$, where $y_1,\ldots,y_N$ are function parameters,

loading a set of functions $g_1,\ldots,g_N$;

applying the execution device function cascade to the set of functions $g_1,\ldots,g_N$; where:

g_i is functionally equivalent to $p_{2i}^{-1} \circ f_i \circ p_{2i-1}$, for $1 \le i \le N$;

h_i is functionally equivalent to $p_{2i-1}^{-1} \circ p_{2i-2}$ for $2 \le i \le N$; and

p_i is an invertible permutation, for $1 \le i \le 2N$.

11. (original) A system for providing a digital signal processing function f to an executing device in an obfuscated form; the system including a server (610) and an executing device (620); the function f including a function cascade including a plurality of signal processing functions f_i, $1 \le i \le N$, for processing a digital signal input x to yield a digital signal output (for example, $FC_1(x) \equiv f_N \circ \cdots \circ f_1(x)$);

the server including a processor (612) for, under control of a program:

selecting a set of $2N$ invertible permutations p_i, $1 \le i \le 2N$;

calculating a set of N functions g_i, where g_i is functionally equivalent to $p_{2i}^{-1} \circ f_i \circ p_{2i-1}$, for $1 \le i \le N$; and

calculating a set of N-1 functions h_i, where h_i is functionally equivalent to $p_{2i-1}^{-1} \circ p_{2i-2}$, for $2 \le i \le N$; and

means (614) for equipping the executing device with an execution device function cascade that includes $y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \ldots \circ y_1$, where $y_1,\ldots,y_N$ are function parameters $\left(\text{for example, } ED_1(y_1,\ldots,y_N) \equiv y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \ldots \circ y_1\right)$, and

means (616) for providing the functions $g_1,\ldots,g_N$ to the executing device; and

the executing device (620) including:

means (626) for obtaining the functions $g_1,\ldots,g_N$ from the server; and

a processor (622) for, under control of a program, loading the execution device function cascade and applying the loaded execution device function cascade to the functions $g_1,\ldots,g_N$ $\left(\text{for example, } ED_1(g_1,\ldots,g_N)\right)$.

12. (original) An execution device (620) for use in the system as claimed in claim 11; the executing device including:

means (626) for obtaining the functions $g_1,\ldots,g_N$ from the server; and

a processor (622) for, under control of a program, applying the execution device function cascade to the functions $g_1,\ldots,g_N$ $\left(\text{for example, } ED_1(g_1,\ldots,g_N)\right)$ and applying the applied device function cascade to the digital signal input x.

13. (original) A method of providing a digital signal processing function f to a plurality of executing devices, each identified by a unique index j, in an obfuscated, anonymous form; the function f including a function cascade including a plurality of signal processing functions f_i, where $1 \leq i \leq N$, for processing a digital

signal input x to yield a digital signal output (for example, $FC_1(x) \equiv f_N \circ \cdots \circ f_1(x)$), the method including:

selecting a set of $2N$ invertible permutations p_i, where $1 \le i \le 2N$;

calculating a set of N functions g_i, where g_i is functionally equivalent to $p_{2i}^{-1} \circ f_i \circ p_{2i-1}$, $1 \le i \le N$;

selecting for each device j a corresponding set and/or sequence of $2N$ invertible permutations $p_{j,i}$, that is unique for the device and/or a user of the device;

calculating for each executing device j a corresponding set of N-1 functions $h_{j,i}$, where $h_{j,i}$ is functionally equivalent to $p_{j,2i-1}^{-1} \circ p_{j,2i-2}$ for $2 \le i \le N$;

equipping each executing device j with a respective execution device function cascade $ED_j(y_1,\ldots,y_N)$ that includes $y_N \circ h_{j,N} \circ y_{N-1} \circ h_{j,N-1} \circ \ldots \circ y_1$;

equipping each executing device j with a respective loader function $loader_j(x_1,\ldots,x_N) = (l_{j,1} \circ x_1 \circ r_{j,1},\ldots,l_{j,N} \circ x_N \circ r_{j,N})$, where $l_{j,i}$ is functionally equivalent to $p_{j,2i}^{-1} \circ p_{2i}$ and $r_{j,i}$ is functionally equivalent to $p_{2i-1}^{-1} \circ p_{j,2i-1}$;

providing to the executing device the functions $g_1,\ldots,g_N$; and

in the executing device, executing $ED_j(loader_j(g_1,\ldots,g_N))$.

14. (original) A method of providing a digital signal processing function f as claimed in claim 13, including providing $g_1,...,g_N$ to each executing device through broadcasting and/or distribution on a storage medium with a same content for each executing device.

15. (original) A method of providing a digital signal processing function f as claimed in claim 14, including also providing the digital signal input x to each executing device through broadcasting and/or distribution on a storage medium with a same content for each executing device.

16. (original) A method of providing a digital signal processing function f as claimed in claim 13, including providing to executing device j through a one-to-one communication channel and/or a storage medium with a device-specific content at least one the following sets of corresponding functions: $h_{j,i}$, $l_{j,i}$, or $r_{j,i}$.

17. (currently amended) A method of providing a digital signal processing function f as claimed in claim 1~~ or 13~~, wherein the function f is a decryption function based on a Feistel cipher network and each of the signal processing functions f_i is a respective Feistel decryption round function.

18. (original) A method of providing a digital signal processing function f as claimed in claim 17, wherein each of the permutations p_i is a Feistel transformer where a function Q operating on a sequential pair <x, y> is a Feistel transformer if there exist invertible functions Q_x and Q_y and $Q(\langle x,y\rangle)=\langle Q_x(x),Q_y(y)\rangle$, where $Q_x(\mathrm{x})\oplus Q_x(y)=Q_x(x\oplus y)$ and $Q_y(\mathrm{x})\oplus Q_y(y)=Q_y(x\oplus y)$

19. (original) A computer program product operative to cause a processor in an execution device j to execute a digital signal processing function f including a function cascade including a plurality of signal processing functions f_i, where $1\le i\le N$, for processing a digital signal input x to yield a digital signal output (for example, $FC_1(x)\equiv f_N\circ\cdots\circ f_1(x)$), the method including:

loading an execution device function cascade that is unique for the execution device and that includes $y_N\circ h_{j,N}\circ y_{N-1}\circ h_{j,N-1}\circ\ldots\circ y_1$, where $y_1,\ldots,y_N$ are function parameters,

loading a loader function $loader_j(x_1,\ldots,x_N)\equiv(l_{j,1}\circ x_1\circ r_{j,1},\ldots,l_{j,N}\circ x_N\circ r_{j,N})$,

loading a set of functions $g_1,\ldots,g_N$;

applying the loader function to the set of functions $g_1,\ldots,g_N$ yielding a set of functions $g_{j,1},\ldots,g_{j,N}$ and applying the

execution device function cascade to the set of functions $g_{j,1}, \ldots,$ $g_{j,N}$,

where:

g_i is functionally equivalent to $p_{2i}^{-1} \circ f_i \circ p_{2i-1}$, for $1 \leq i \leq N$;

p_i is an invertible permutation, for $1 \leq i \leq N$;

$h_{j,i}$ is functionally equivalent to $p_{j,2i-1}^{-1} \circ p_{j,2i-2}$ for $2 \leq i \leq N$;

$l_{j,i}$ is functionally equivalent to $p_{j,2i}^{-1} \circ p_{2i}$;

$r_{j,i}$ is functionally equivalent to $p_{2i-1}{}^{-1} \circ p_{j,2i-1}$; and

$p_{j,i}$ are invertible permutations, for $1 \leq i \leq 2N$, being unique for the device and/or a user of the device.

20. (original) A system for providing a digital signal processing function f to a plurality of executing devices, in an obfuscated, anonymous form; the system including a server and a plurality of executing devices, each identified by a unique index j; the function f including a function cascade including a plurality of signal processing functions f_i, where $1 \leq i \leq N$, for processing a digital signal input x to yield a digital signal output (for example, $(FC_1(x) \equiv f_N \circ \cdots \circ f_1(x))$;

the server including a processor for, under control of a program:

selecting a set of $2N$ invertible permutations p_i, where $1 \le i \le 2N$;

calculating a set of N functions g_i, where g_i is functionally equivalent to $p_{2i}^{-1} \circ f_i \circ p_{2i-1}$, for $1 \le i \le N$;

selecting for each device j a corresponding set and/or sequence of $2N$ invertible permutations $p_{j,i}$, that is unique for the device and/or a user of the device;

calculating for each executing device j a corresponding set of N-1 functions $h_{j,i}$, where $h_{j,i}$ is functionally equivalent to $p_{j,2i-1}^{-1} \circ p_{j,2i-2}$ for $2 \le i \le N$;

equipping each executing device j with a respective execution device function cascade $ED_j(y_1,...,y_N)$ that includes $y_N \circ h_{j,N} \circ y_{N-1} \circ h_{j,N-1} \circ ... \circ y_1$;

equipping each executing device j with a respective loader function $loader_j(x_1,...,x_N) = (l_{j,1} \circ x_1 \circ r_{j,1},...,l_{j,N} \circ x_N \circ r_{j,N})$, where $l_{j,i}$ is functionally equivalent to $p_{j,2i}^{-1} \circ p_{2i}$ and $r_{j,i}$ is functionally equivalent to $p_{2i-1}^{-1} \circ p_{j,2i-1}$; and

providing to the executing device the functions $g_1,...,g_N$; and

each executing device j,

means for obtaining the functions $g_1,...,g_N$ from the server; and

a processor for, under control of a program:

loading an execution device function cascade that is unique for the execution device and that includes $y_N \circ h_{j,N} \circ y_{N-1} \circ h_{j,N-1} \circ \ldots \circ y_1$, where $y_1,\ldots,y_N$ are function parameters,

loading a loader function

$loader_j(x_1,\ldots,x_N) \equiv (l_{j,1} \circ x_1 \circ r_{j,1},\ldots,l_{j,N} \circ x_N \circ r_{j,N})$,

applying the loader function to the set of functions $g_1, \ldots, g_N$ yielding a set of functions $g_{j,1},\ldots,g_{j,N}$; and

applying the execution device function cascade to the set of functions $g_{j,1},\ldots,g_{j,N}$

21. (original) An execution device for use in the system as claimed in claim 20; where the executing device is identified by a unique index j; and includes:

means for obtaining the functions $g_1,\ldots,g_N$ from the server; and

a processor for, under control of a program:

loading an execution device function cascade that is unique for the execution device and that includes $y_N \circ h_{j,N} \circ y_{N-1} \circ h_{j,N-1} \circ \ldots \circ y_1$, where $y_1,\ldots,y_N$ are function parameters,

loading a loader function

$loader_j(x_1,\ldots,x_N) \equiv (l_{j,1} \circ x_1 \circ r_{j,1},\ldots,l_{j,N} \circ x_N \circ r_{j,N})$,

applying the loader function to the set of functions $g_1,\ldots,g_N$ yielding a set of functions $g_{j,1},\ldots,g_{j,N}$; and

applying the execution device function cascade to the set of functions $g_{j,1},\ldots,g_{j,N}$,

where:

g_i is functionally equivalent to $p_{2i}^{-1} \circ f_i \circ p_{2i-1}$, for $1 \le i \le N$;

p_i is an invertible permutation, for $1 \le i \le N$;

$h_{j,i}$ is functionally equivalent to $p_{j,2i-1}^{-1} \circ p_{j,2i-2}$ for $2 \le i \le N$;

$l_{j,i}$ is functionally equivalent to $p_{j,2i}^{-1} \circ p_{2i}$;

$r_{j,i}$ is functionally equivalent to $p_{2i-1}^{-1} \circ p_{j,2i-1}$; and

$p_{j,i}$ are invertible permutations, for $1 \le i \le 2N$, being unique for the device and/or a user of the device.